

## Selection of relay nodes in Bluetooth Mesh network for smart home application

Phan Van Hai, Nguyen Quoc Cuong\*

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam.

\*Corresponding author: cuong.nguyenquoc@hust.edu.vn

Received: 03 May 2023; Revised 20 Nov. 2023; Accepted 10 May 2024; Published 20 May 2024.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.95.2024.1-11>

### ABSTRACT

*Bluetooth Mesh (BM) is increasingly growing and appearing in Internet of Things (IoT) systems. BM uses a managed flooding mechanism to transmit the message. Nodes with a relay feature will receive and forward the message to neighboring nodes, so that the message can be spread to the whole network. The default configuration of BM is that all nodes are relays, ensuring that every node in the network can connect to each other. However, when all nodes act as relays, it results in a lot of messages being transmitted through the network at a time, which increases the possibility of packet collisions, thereby reducing the packet delivery rate (PDR) and increasing the end-to-end delay (E2E delay). These issues will greatly affect the user experience, especially in smart home systems where users interact near the device. Therefore, it is necessary to choose the relay nodes in the network appropriately to improve the quality of the network. This paper will select and analyze some relay node selection algorithms in the BM network applied in the smart home system, and compare them with the default configuration of the network (all relays) in terms of PDR and E2E delay criteria.*

**Keywords:** Bluetooth mesh; Smarthome; Routing algorithm.

### 1. INTRODUCTION

Bluetooth Mesh [1] (BM) is growing in popularity in IoT (Internet of Things) systems, particularly in lighting and smart home applications. In these systems, users frequently interact near devices, so the system must ensure quality to provide a positive user experience. Otherwise, the user will feel very uncomfortable when using it and, in the worst case, may not use the system anymore. The user experience is directly impacted by the packet delivery rate (PDR) and end-to-end delay (E2E Delay). PDR can simply be understood as the rate of successful sending of a message to the target device. The user always wants this rate to reach 100%. They don't want to presume that the control shuts off 10 lights when in fact it only turns off 9 lights and one light stays on. E2E delay can simply be understood as the delay time from the time a source device sends a message to the time the destination device receives that message. If the user controls a device where the network has a lot of delay, the user may think that it did not receive the request and will make many more requests, this will make the user feel uncomfortable and also affect to the system when having to handle many requests sent continuously.

The nodes in the Bluetooth Mesh network are divided into many types depending on the features, in which the relay node type is the default configuration. The relay node is the node that will forward the messages it receives to the neighborhood nodes, helping the message reach its destination. If a BM network has all nodes as relay nodes, it can lead to PDR and E2E delay problems. When all nodes are relayed, each transmitted message is forwarded by many nodes in the network, and when many messages are

transmitted, a lot of messages are forwarded in the network at a time, which causes packet collision and lowers PDR. In addition, increased packet conflicts can also lead to increased E2E delay. Suppose a source node sends a message to a destination node, and the message can take many paths to reach the destination, including the shortest path called *MinPath*. If the message follows this *MinPath* path, the delay will be minimal. But if the packet conflicts happen a lot, leading to *MinPath* may not be successful, the message will have to take another route to reach the destination. This means E2E delay will increase. Therefore, with all nodes configured as relays, packet collisions will occur a lot, leading to reduced PDR and increased E2E delay. But if the network has very few relay nodes, it will also adversely affect the PDR. Because when there are too few relay nodes, messages will have fewer paths to reach their destination, while if the number of messages to be sent is large, it will clog the network and overload it, leading to message loss. This means that the PDR will be low. Therefore, the selection of the relay node has a direct effect on the PDR and E2E delay.

So far, there are very few studies on the selection of relay nodes in BM networks, and in smart home applications, there are no studies. Therefore, the objective of this study is to search, analyze and evaluate some relay node selection algorithms to improve PDR and E2E delay applied in smart home systems. These reviews will be useful references in finding a configuration algorithm to improve the quality of the Bluetooth Mesh network. The rest of the paper will be organized as follows: Section 2 will present an overview of Bluetooth Mesh networks; Section 3 will present the relay selection algorithms that will be used for evaluation; Section 4 will present simulation parameters and scenarios; Section 5 presents simulation results and evaluation results; and section 6 is the conclusion.

## 2. BLUETOOTH MESH OVERVIEW

The Bluetooth Mesh [1] has a layered architecture built on the basis of the Bluetooth Low Energy protocol [2]. The layers of the Bluetooth Mesh from top to bottom include the models layer, foundation models, access layer, upper transport layer, lower transport layer, network layer, and bearer layer.

The model for exchanging messages between nodes is the publish-subscribe model. Nodes can generate and publish messages to unicast, group, or virtual addresses. Nodes that want to receive these messages will subscribe to that address. Some special and fixed group addresses, for example, address 0xFFFF which points to all nodes. If a node sends a message to a group address of 0xFFFF, all other nodes can receive and process that message. All nodes in the network can transmit and receive messages, and nodes can optionally support one or more other features. There are four optional features: relay, proxy, friend, and low power. The relay feature allows the node to receive and retransmit messages, making it possible for the message to be sent to nodes far away from the transmitter. The Proxy feature lets the node receive and transmit messages between GATT and advertising bearers. The Low Power feature allows the node to operate with low power consumption by linking to a node with the Friend feature. The Friend feature allows the node to support low-power nodes by storing incoming messages for the associated low-power nodes.

BM uses a technology called "managed flooding" to deliver messages. Accordingly, nodes in the network transmit messages by broadcasting them to neighboring nodes. If

they receive the message and are relay nodes, these neighbors will also forward the message by broadcasting it to their neighbors. Using relay nodes, this will extend the transmission distance of the message. Any node can send a message to other nodes outside its transmission range if there are relay nodes between them. Managed flooding also offers several ways to limit and control message forwarding, avoiding messages being forwarded and repeated indefinitely. The two main methods are the network message cache and time-to-live (TTL) methods. The network cache prevents the re-forwarding of previously received messages by saving messages to a checklist. Each message has a TTL value to limit the number of times it can be forwarded. Each time a message is received and forwarded (up to 126 times), the TTL is reduced by 1. If the TTL is less than or equal to 1, the message will not be forwarded anymore.

Managed flooding technology has the advantage of being simple, without the need for complicated routing in message delivery. In particular, the relay node plays a very important role so that all nodes in the network can be linked together, building up a mesh network. By default, all nodes in the network are relay nodes, ensuring that every node is always connected. However, this default configuration can reduce the quality of the network and the user experience, particularly by reducing the PDR and increasing the E2E delay, as analyzed in the previous section. Therefore, choosing a suitable set of relay nodes is necessary to improve the quality of the Bluetooth network.

### **3. RELAY SELECTION ALGORITHMS**

The selection of relay nodes in the BM network is a problem of finding the Connected Dominating Set (CDS) in wireless sensor networks (WSNs), in which the CDS is the set of nodes that will act as relays. In WSNs, assuming nodes have the same transmission-receiving distance, two nodes are neighbors if they are in each other's transmit-receive areas. A Dominating Set (DS) is defined as a subset of a network in which each node is either a member of the DS or a neighbor of a node in the DS. If the graph obtained by the nodes in the DS set is connected, then this DS set is called CDS [3]. CDS ensures the entire network is connected, i.e., any two nodes can be connected through intermediate nodes in CDS.

Algorithms for finding CDS can be divided into two categories: centralized and decentralized (including local and distributed). In a centralized algorithm, the processing and computation are performed at a place such as a gateway or a server based on the information of the entire network. Meanwhile, the processing and computation of the decentralized algorithm are performed at the nodes in the network based on the information surrounding each node. Each type of algorithm has its advantages and disadvantages. The centralized algorithm has some advantages, such as being processed quickly and efficiently because the processing equipment often has a large computing capacity; updating or changing the algorithm is also easy because it only needs to be changed at the processing place.

In a smart home system using BM, there is usually a central control device called a gateway with an internet connection to communicate with the server. In addition, it can communicate with devices in the BM network. In other words, the BM will be centrally managed and controlled by the gateway device. With this feature, it is easy to apply centralized algorithms to the system. Therefore, the research will focus on relay node selection algorithms based on the problem of finding a centralized CDS.

So far, much research has yet to be done on selecting relay nodes in BM networks. The study [4] proposed three distributed algorithms: Greedy Connect, K2 Pruning, and Dominator. This study evaluates fault tolerance characteristics such as the number of relay nodes in the network, the number of nodes that lose connection when removing a relay node, and how many relay nodes can be removed before the network is disconnected. In addition, the study also focuses on evaluating PDR in networks with different node densities. The study [5] compares several relay node selection algorithms, including three distributed algorithms (A1, A2, A3), one local (A4), and one centralized (A5). This study tested a fixed-area network with different scenarios in terms of the number of nodes and compared them based on a variety of criteria, such as PDR, E2E delay, collision, and energy savings.

The algorithms used in the two studies are also CDS find algorithms. According to the study's results [4], all three algorithms have improved PDR compared to the default algorithm (all nodes are relay nodes), but not considering E2ED, and all three algorithms are not centralized. According to the study's results [5], all five algorithms did not improve PDR and E2ED compared to the default algorithm, in which the centralized algorithm (A5) had the worst results. Therefore, this study will find a different centralized CDS finding algorithm with the desire to improve PDR and E2ED.

There are many algorithms to find CDS in WSN, in which the study chooses two centralized algorithms, GOC-MCDS-C [3] and (1, k)-CDS [6]. Both algorithms follow two steps: the first is to find the sets of maximal independent sets (MIS); the second is to select other nodes to connect the MIS to satisfy the algorithm's goal. The nodes in the MIS and the nodes selected in Step 2 form the CDS. Let  $G$  be a graph with a vertex set  $V$ . A subset  $U \subseteq V$  is said to be an independent set of  $G$  if any two nodes in  $U$  are not neighbors, and if each vertex in the set  $V \setminus U$  is a neighborhood of at least one vertex in  $U$ , then  $U$  is called MIS [6]. When applying these two algorithms together with some small changes, the study presents two modified algorithms: T1 and T2. For both algorithms T1 and T2, we will use a simple MIS algorithm (presented in study [7]), which is referred to in this paper as the SimpleMIS algorithm.

### 3.1. Algorithm T1

The algorithm T1 is based on the GOC-MCDS-C algorithm, proposed by Hongwei Du et al. in [3]. GOC-MCDS-C considers the minimization of the CDS set and the reduction of the routing cost. The algorithm will produce a CDS of limited size within a constant factor. In addition, for any two nodes  $u$  and  $v$ , there will exist a path with intermediate nodes located in the CDS whose path length is at most  $7 * d(u, v)$ , where  $d(u, v)$  is the shortest path length of  $u$  and  $v$ . The algorithm makes the path between any two nodes not too long. However, the shorter the path, the smaller the E2E delay, so logically, this algorithm will help the E2E delay of the network be small. That is why we chose this algorithm to apply to the BM network with the expectation of getting a network with a small E2E delay.

The algorithm can be divided into two steps:

- Step 1: Construct MIS  $I$ ;
- Step 2: For each pair of nodes  $u, v$  in set  $I$  with  $d(u, v) \leq 3$ , find the shortest

path  $p(u, v)$  and put all nodes intermediate of that path into set  $C$ .  $CDS = I \cup C$  is the result set of the algorithm. However, GOC-MCDS-C did not specify which algorithm to use to find MIS. So in the algorithm T1, we use the SimpleMIS algorithm to find MIS.

Below is the pseudocode for algorithm T1.

---

**Algorithm T1**

---

**Input:** Undirected graph  $G(V, E)$

**Output:** CDS  $D$  of graph  $G$

```

1   $D = \emptyset$ 
2   $I = \text{SimpleMIS}(G)$ 
3  for  $\forall u, v \in I$  do
4      if  $d(u, v) \leq 3$  then
5           $P(u, v) = \text{Set of nodes on a shortest path between } u \text{ and } v$ 
6           $D = D \cup (P(u, v) \setminus \{u, v\})$ 
7      end if
8  end for
9  Return  $D = D \cup I$ 

```

---

**3.2. Algorithm T2**

T2 is based on the (1, k)-CDS algorithm in [6]. (1, k)-CDS, which aimed at solving the problem of finding the minimum k-tuple dominating set. Let  $G$  be a graph with a vertex set  $V$ , a k-tuple dominating set  $S \subseteq V$  of  $G$  is a set of vertices such that each vertex  $u \in V$  is k-dominated by vertices of  $S$ .  $k \leq \delta(G)$  where  $\delta(G)$  is the minimum degree of vertices in  $V(G)$ . This algorithm considers not only CDS minimization but also balance efficiency and fault tolerance. The algorithm makes the network fault-tolerant by ensuring that each node will have multiple neighbors in the CDS, i.e., ensure each node will have multiple paths to it. This feature allows a message to reach its destination in many different paths. Logically, the algorithm increases the probability that the message reaches the destination successfully or increases the PDR. That is why we chose this algorithm to apply to BM, expecting to obtain a network with a large PDR.

The algorithm can be divided into two steps:

- Step 1: Construct a set  $D$  by constructing  $k$  times MIS so that every vertex in  $V \setminus D$  is k-dominated by set  $D$ ;
  - Step 2: Select from set  $V \setminus D$  the nodes to put into set  $D$  such that every vertex in set  $D$  set in Step 1 is also k-dominated. The resulting set  $D$  can have 2 cases: If  $D$  is a connected set, then the resulting CDS set is  $D$ ; otherwise, another algorithm needs to be used to connect set  $D$  and form the resulting CDS set. The algorithm used to connect set  $D$  in this case will be the T1 algorithm. T2 is the same as (1, k)-CDS, but with some changes, mainly in the algorithm to find the MIS.
-

Specifically, T2 uses the SimpleMIS algorithm instead of the MIS find algorithm suggested by (1, k)-CDS.

Below is the pseudocode for algorithm T2.

---

**Algorithm T2**

---

**Input:** Undirected graph  $G(V, E)$ , coefficient  $k$

**Output:** CDS  $D$  of graph  $G$

```

1   $G' = G$ 
2  for  $i = 1$  to  $k$  do
3       $I_i = \text{SimpleMIS}(G')$ 
4       $G' = G' \setminus (I_1 \cup \dots \cup I_{i-1})$ 
5  end for
6   $D_k = I_1 \cup \dots \cup I_k$ 
7   $D = D_k$ 
8  The missing dominator number of  $v$ :  $m(v) = k - |N(v) \cap D| \forall v \in D_k$ 
   ( $N(v)$  is the set of neighboring nodes of  $v$ )
9  Set of nodes not having enough  $k$  dominating nodes:  $S$ 
    $= \{v \in D_k \mid m(v) > 0\}$ 
10 while  $S \neq \emptyset$  do
11     Find  $u \in V \setminus D$  such that  $u$  has the most neighbors in  $S$ 
12      $D = D \cup \{u\}$ 
13      $m(v) = m(v) - 1 \forall v \in N(u) \cap S$ 
14      $S = \{v \in D_k \mid m(v) > 0\}$ 
15 end while
16 if  $D$  is not connected then
17      $C = T1(G)$ 
18      $D = D \cup C$ 
19 end if
20 Return  $D$ 

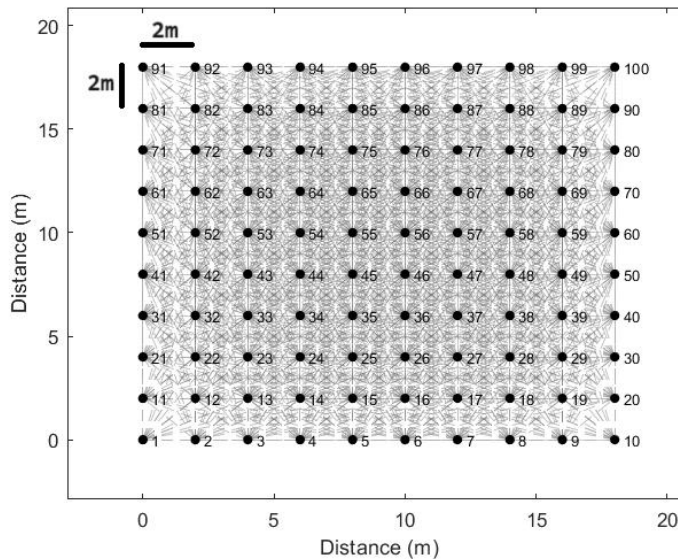
```

---

#### 4. SIMULATION SCENARIO AND METRICS

As mentioned in the previous section, a smart home system using BM usually has a central control device called a gateway. The gateway will send requests to actuators in the network, such as lights and fans, and receive status information from actuators or sensors sent back. Smart homes usually have a small area, and the nodes in the system are often placed close to each other. For example, in the lighting system, they are often placed in

the form of a grid and are about 2 m apart. Therefore, this paper will simulate a BM network on a floor area of 18 m × 18 m. The nodes are arranged in a grid, as shown in figure 1. We will only consider the types of nodes as relay nodes and end nodes (which do not support relay, friend, low power, and proxy features). The network has four message transmitter nodes, of which one will act as a gateway. It will send messages to every node in the network (by sending to address 0xFFFF). The remaining three nodes act as sensors, sending messages to the gateway node, the other nodes in the network act as actuators that only receive the messages. The message type requires no response, is not segmented, and has a default TTL of 127. We perform two different simulation scenarios for two different locations of the gateway node. In scenario 1, the gateway node is node 1. In scenario 2, the gateway node is node 45. The three fixed sensor nodes in the two scenarios are 10, 91, and 100. The three sensors are placed at the corners of the house to simulate real-world sensors like door locks and door sensors. The position of the gateway in scenario 1 is chosen to test the case where the gateway is farthest from the sensors. In scenario 2, the gateway is positioned to simulate real-world situations where the gateway is typically placed in the middle of the house. For the simulation network, the minimum order of vertices is  $\delta(G) = 21$ . Therefore, for each scenario, we will run the algorithms, including T1 and T2 with k from 1 to 21, and the default algorithm of the network, i.e., every node is a relay. For each algorithm, we run the simulation ten times, each time differing in the starting time of the four nodes. All four nodes will not transmit at the beginning of the simulation, but they will start transmitting at different times to create asynchronous transmission, making the assessment more objective. Specifically, we generate random times in the range of 200 ms for each node.



**Figure 1.** Network layout on one floor in the smart home.

The tool we use is MATLAB. Let  $N$  be the total number of nodes in the network,  $N(i)$  is the  $i$ -th node, and  $N(i = iGW)$  is the gateway node. When we run an algorithm simulation for the  $i$ sum-th time,  $i$ sim = 1,2, ...,  $N$ sim and  $N$ sim = 10 , we will collect the parameters:  $recv(i)$  is the total number of messages received by the  $i$ -th node;  $trans(i)$  is the total number of messages transmitted to the  $i$ -th node;  $e2ed(i)$  is the total

delay of messages received by the  $i$ -th node. From these parameters, we will calculate the parameters:  $PDR_{isim}$  is the average PDR value of the network at  $isim$ -th simulation, and  $E2ED_{isim}$  is the average E2E delay of messages sent by the gateway to the entire network at  $isim$ -th simulation.

$$PDR_{isim} = \frac{\sum_{i=1}^N recv(i)}{\sum_{i=1}^N trans(i)} \quad (1)$$

$$E2ED_{isim} = \frac{\sum_{i=1, i \neq iGW}^N \frac{e2ed(i)}{recv(i)}}{N - 1} \quad (2)$$

Where,  $N - 1$  is the total number of nodes that are not gateway node.

From the  $PDR_i$  and  $E2E\ delay_i$  parameters, we obtain two metrics:

$$PDR_{avg} = \frac{\sum_{i=1}^{Nsum} PDR_i}{Nsim} \quad (3)$$

$$E2ED_{avg} = \frac{\sum_{i=1}^{Nsum} E2E\ delay_i}{Nsim} \quad (4)$$

These two metrics will be used to evaluate the algorithms.  $PDR_{avg}$  is the average packet delivery rate of the network, and we use it to evaluate the packet delivery rate in the network. We are also interested in the latency of the control messages that the user sends to the device, so that we will calculate  $E2ED_{avg}$  as the average delay of the messages sent by the gateway node to the entire network. In addition, we also calculate the standard deviation of each parameter to provide more information when evaluating. Formula to calculate standard deviation  $s$  of data set  $x$  (here PDR and E2ED) with  $N$  elements:

$$s = \sqrt{\frac{1}{N - 1} \sum_{i=1}^N (x_i - \hat{x})^2} \quad (5)$$

in there:

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (6)$$

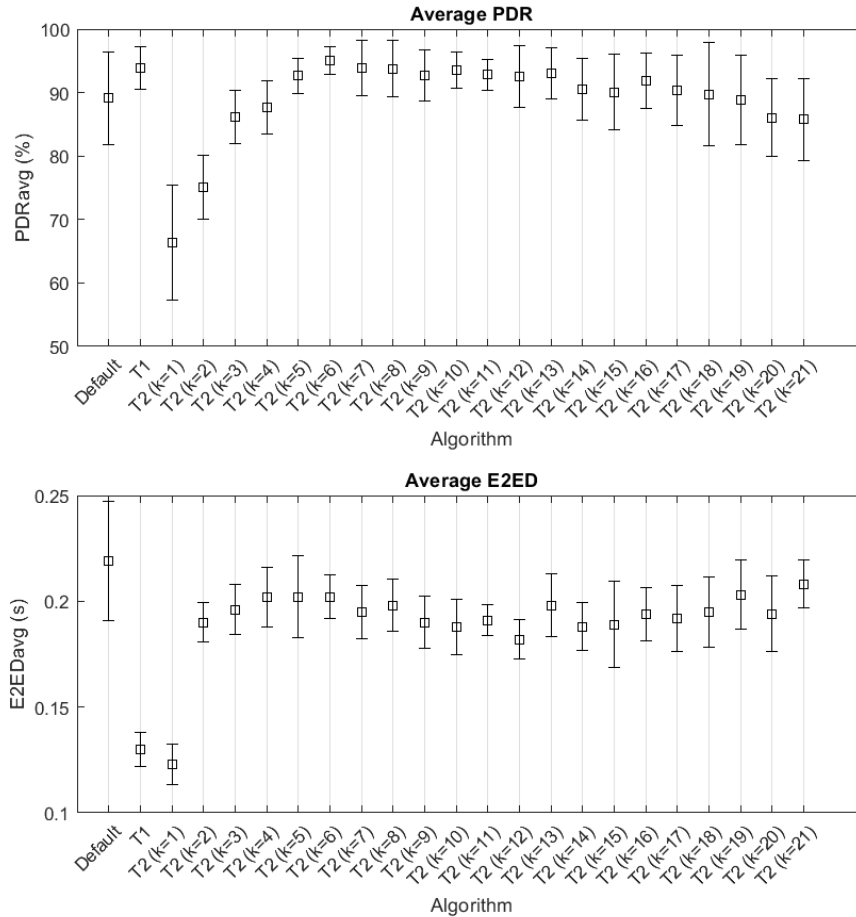
A summary of some of the parameters used in the simulation is shown in table 1.

**Table 1.** Simulation parameters.

The total nodes	100
The transmit-receive distance	9 m
Source-destination node pairs: scenario 1	1-all nodes; 10-1; 91-1; 100-1
Source-destination node pairs: scenario 2	45-all nodes; 10-45; 91-45; 100-45
Message generation rate	4 messages/second
The number of messages sent by each node	32
TTL	127

Advertising interval	20 ms
Scan interval	30 ms
Considering attenuation and obstacles	No

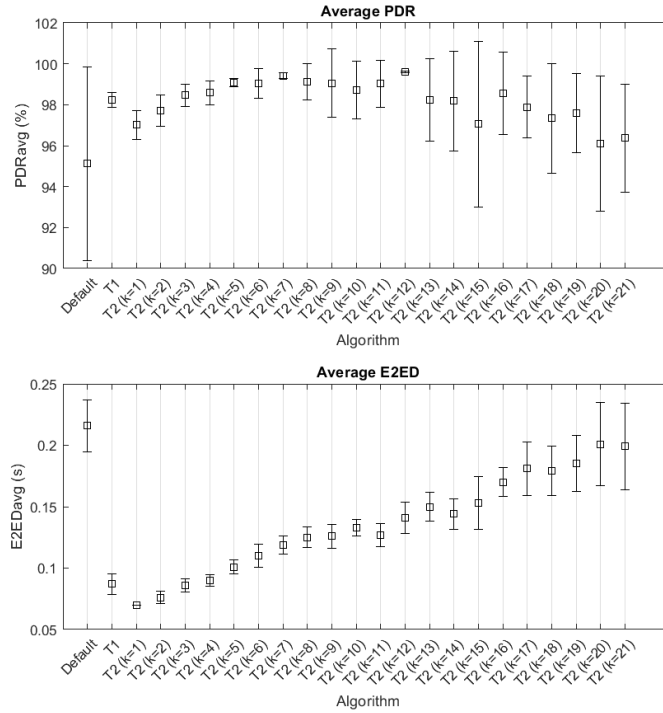
### 5. SIMULATION AND RESULTS



**Figure 2.** Average PDR and average E2E delay in scenario 1.

Figure 2 is the result of scenario 1. Accordingly, the two algorithms T2 (with some values of  $k$ ) and T1 both give better results than the default algorithm of the BM network (all relays), both in terms of PDR and E2E delay. The T2 algorithm with  $k = 6$  gives the best PDR, which is  $95.02 - 89.1 = 5.92\%$  higher than the default algorithm, while the T1 is  $93.92 - 89.1 = 4.82\%$  higher. Meanwhile, in terms of E2E delay, the T1 algorithm has a much lower delay than T2 (with  $k \geq 2$ ) and than the default algorithm. With  $k = 1$ , T2 has lower delay than T1, but the PDR is too low.

Figure 3 is the result of scenario 2. Accordingly, the two algorithms T2 (with some values of  $k$ ) and T1 still give better results than the default algorithm, both in terms of PDR and E2E delay. The T2 algorithm (with  $k = 12$ ) gives the best PDR,  $99.6 - 95.12 = 4.48\%$  higher than the default algorithm, and the T1 algorithm is  $98.24 - 95.12 = 4.12\%$  higher. Meanwhile, in terms of E2E delay, the T1 algorithm still has a much lower delay than T2 when  $k \geq 4$  and compared to the default algorithm.



**Figure 3.** Average PDR and average E2E delay in scenario 2.

From the above results, we have the following conclusions: both T2 algorithms (at some values of  $k$ ) and T1 improve the overall performance of the network in terms of PDR and E2E delay better than the default network configuration. The results meet the expectations that we stated in the previous section when analyzing the characteristics of the two algorithms. In which T2 is more inclined to increase PDR, T2 is more inclined to reduce E2E delay. In addition, it should be noted that the value of  $k$  for T2 to have the best PDR is not fixed, and the choice of  $k$  value for each different network is an unsolved problem. Therefore, we recommend choosing the T1 algorithm because the simple deployment algorithm is suitable for deploying self-configured network systems without adjustment coefficients like T2 and, more importantly, is an improvement in both PDR and End-to-End delay.

As previously mentioned, there haven't been many studies related to the relay node selection algorithm in BM networks, making it challenging to compare these two algorithms with others. However, in the mentioned study [5], the algorithms presented did not show improvement in PDR and E2ED. Therefore, in a qualitative comparison, algorithms T1 and T2 might be better than those in [5].

## 6. CONCLUSIONS

This paper evaluates and compares relay node selection algorithms to improve the overall quality of the BM network in the smart home system in terms of PDR and end-to-end delay. The two algorithms we chose, T1 is based on the GOC-MCDS-C algorithm and T2 is based on the  $(1, k)$ -CDS algorithm, both help increase PDR and reduce end-to-end delay compared to the network using the default configuration (all nodes are relays). In which, T1 helps to reduce delay more than T2, but in terms of PDR, T2 at some  $k$  values

will be higher than T1. However, for the T2 algorithm, the value of k to get the best PDR is not fixed, so choosing k for each network is a difficult problem. Therefore, the T1 algorithm will be more suitable for finding the BM network configuration algorithm to improve the PDR and E2E delay of the entire network.

## REFERENCES

- [1]. Bluetooth SIG, “*Mesh Profile 1.0.1.*” (2019). [Online]. Available: <https://www.bluetooth.com/specifications/specs/mesh-profile-1-0-1/>
- [2]. Bluetooth SIG, “*Core Specification 5.4.*” (2023). [Online]. Available: <https://www.bluetooth.com/specifications/specs/core-specification-5-4/>
- [3]. Hongwei Du, Weili Wu, Qiang Ye, Deying Li, Wonjun Lee, and Xuepeng Xu, “*CDS-Based Virtual Backbone Construction with Guaranteed Routing Cost in Wireless Sensor Networks,*” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 652–661, (2013), doi: 10.1109/TPDS.2012.177.
- [4]. E. A. J. Hansen, M. H. Nielsen, D. E. Serup, R. J. Williams, T. K. Madsen, and R. Abildgren, “*On Relay Selection Approaches in Bluetooth Mesh Networks,*” in 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia: IEEE, pp. 1–5, (2018). doi: 10.1109/ICUMT.2018.8631214.
- [5]. M. Reno *et al.*, “*Relay Node Selection in Bluetooth Mesh Networks,*” in 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy: IEEE, pp. 175–180, (2020). doi: 10.1109/MELECON48756.2020.9140584.
- [6]. W. Shang, P. Wan, F. Yao, and X. Hu, “*Algorithms for minimum m-connected k-tuple dominating set problem,*” *Theor. Comput. Sci.*, vol. 381, no. 1–3, pp. 241–247, (2007), doi: 10.1016/j.tcs.2007.04.035.
- [7]. S. Assadi and S. Solomon, “*When Algorithms for Maximal Independent Set and Maximal Matching Run in Sublinear-Time.*” arXiv, (2020). doi: 10.48550/arXiv.2006.07628.

## TÓM TẮT

### Lựa chọn nút chuyển tiếp trong mạng Bluetooth Mesh cho ứng dụng nhà thông minh

*Bluetooth Mesh (BM) đang ngày càng phát triển và xuất hiện nhiều trong các hệ thống Internet vạn vật (IoT). BM sử dụng cơ chế ngập lụt được quản lý để truyền tin nhắn. Các nút với tính năng chuyển tiếp sẽ nhận và chuyển tiếp tin nhắn tới các nút lân cận, giúp cho tin nhắn có thể lan truyền tới toàn mạng. Cấu hình mặc định của BM là tất cả các nút đều là nút chuyển tiếp, điều này đảm bảo mọi nút trong mạng có thể kết nối được với nhau. Tuy nhiên, khi tất cả các nút đóng vai trò là nút chuyển tiếp, sẽ dẫn đến có rất nhiều tin nhắn được truyền đi trong mạng ở cùng một thời điểm, điều này sẽ tăng khả năng xảy ra xung đột gói tin, từ đó dẫn đến giảm tỉ lệ chuyển giao gói tin (PDR) và tăng độ trễ điểm-tới-điểm (E2E delay). Những vấn đề này sẽ ảnh hưởng lớn tới trải nghiệm người dùng, đặc biệt trong các hệ thống nhà thông minh nơi mà người dùng thường xuyên tương tác ở gần thiết bị. Do vậy, cần lựa chọn các nút chuyển tiếp trong mạng một cách phù hợp để đảm bảo cải thiện chất lượng của mạng. Bài báo này sẽ lựa chọn và phân tích một số thuật toán lựa chọn nút chuyển tiếp trong mạng BM ứng dụng trong hệ thống nhà thông minh, và so sánh các thuật toán đó với cấu hình mặc định của mạng (tất cả nút là nút chuyển tiếp) về tiêu chí PDR và E2E delay.*

**Từ khoá:** Bluetooth mesh; Nhà thông minh; Thuật toán định tuyến.